3.0 SUMMARY OF ASSUMPTIONS, LIMITATIONS, AND ERRORS

This section describes the assumptions, limitations, and known errors associated with SWEG, as well as the implications for model use for each assumption, limitation, and error listed. This information is essential in helping a user to determine if the model adequately addresses all the phenomena and environmental conditions that are important to the intended application.

The information in this section is based on SWEG version 6.5.5 which was released in January 1997.

3.1 ASSUMPTIONS

Table 3-1 lists important model assumptions in the implementation of particular functions within SWEG. For each assumption listed there is a brief assessment of implications for model use. The assumptions listed here are at a high level; that is, they are applicable to broad functional areas such as platform movement and target detection. Additional assumptions pertaining to specific functional areas will be included in Section 2 of ASP-II as verification efforts are completed. The assumptions, limitations, and errors are categorized according to the Functional Area Template (FAT) found in Appendix A.

TABLE 3-1. Assumptions for SWEG 6.5.5 by Functional Element.

Functional Element	Assumption	Implications for Model Use
General	• SWEG will be able to interface with other assets (simulations, simulators, hardware or people in the loop, etc.) in a networked environment which may or may not be real-time.	• The model runs quickly and cannot go backward in time.
	• When used with other assets, SWEG will receive accurate information about each of them.	• The user must provide instructions with links to all assets and accurate initial information about them. Other assets must be able to provide accurate information about themselves.
	• When used with other assets, SWEG, by default, will represent PLAYERs not explicitly delegated to other assets.	• The user must provide explicit instructions for any PLAYERs to be represented by other assets.
	• All PLAYER activities can be categorized as one of six generic functions: moving, shooting, communicating, sensing, disrupting, and thinking.	• Users must frame instructions within these categories.
	• PLATFORMS leave the exercise without leaving any physical evidence.	• The simulation does not explicitly include the effects of the hulks of destroyed PLATFORMs on the activities of other PLATFORMs.

TABLE 3-1. Assumptions for SWEG 6.5.5 by Functional Element. (Contd.)

Functional Element	**Assumption**	**Implications for Model Use**
Platform 1.2 Movement	• Space and time are represented by Newtonian physics.	• Time dilation, space contraction, and other relativistic effects are not explicitly simulated.
	• Mass is not explicitly represented.	• If desired, the user must implicitly represent the effects of mass in data such as turn limits, climb and dive limits, etc.
	• Movement paths are represented by a series of straight line segments and arcs of circles.	• The fidelity of simulated movement is limited only by user instructions.
Platform 2.0 Sensors	• Energy transmission is instantaneous.	• Time required for detection may be unrealistically short, especially for acoustic sensors.
	• The level of detail in sensor-derived perceptions is the same as the level of detail that was defined by the user in the perceived platform.	• Sensor resolution is perfectly accurate in the context of the simulation.
	• Locations of detected targets are accurately perceived.	• Location information derived from sensors may be unrealistically accurate.
	• Signal polarization does not affect detection.	• Detections may occur for mismatched emitters and receivers, this may be important for warning receivers.
	• Target fluctuation and signal integration is not explicitly modeled.	• Detection results may be unrealistic.
	• IFF is performed at each detection if a PLAYER has IFF capability.	• This is not the usual procedure in world reality.
	• IFF classification of a target depends on only the SIDEs of the sensor and the target.	• Since both platforms could belong to more than one SIDE, IFF results may not have the intended meaning, and may not be useful.
Platform 4.0 Comm Devices	• Energy transmission is instantaneous.	• This may lead to unrealistically short communication times, especially for acoustic communications.
	• Signal polarization does not affect message reception.	• Polarization mismatch between transmitters and receivers does not affect message reception.
Platform 5.0 CM/CCM	• Energy transmission is instantaneous.	• Time required for disruption may be unrealistically short, especially for acoustic jammers.

TABLE 3-1. Assumptions for SWEG 6.5.5 by Functional Element. (Contd.)

Functional Element	**Assumption**	**Implications for Model Use**
Platform 6.2 Knowledge Base	• Knowledge is the result of one of the following: Initial user instructions, Dynamic updates from sensor results, Dynamic updates from messages, or Implicit updates for self and other PLAYERs used as resources.	
	• Each PLAYER with a thinker has its own perceptions, which may be different from the simulation reality.	
	• Perceptions are originally derived from model reality.	• False targets and phantasms are not explicitly represented; however, it is possible to simulate these via definitions of unusual PLAYERS.
	• Perceptions that are different from model reality are the result of user instructions or time lag since the last update.	• Time lags do not automatically cause inaccurate perceptions (thinkers can extrapolate movement paths), but they may cause these if the perceived mover has changed speed or direction since the last update.
	• Friendly perceptions are aggregated at the PLAYER level of detail; i.e., unless explicitly detected, the entire PLAYER is perceived to be at the location of one of its PLATFORMs. (Friendly perceptions are those so defined by the user and those which are implicitly obtained whenever a PLAYER evaluates a TACTICAL CRITERION with an associated resource that is not part of the PLAYER. In the second case, kinematic attributes of the resource are implicitly received at the evaluation time; this information is accurate for one of the PLATFORMs of the resource PLAYER.)	• This can lead to unwanted decisions if the PLAYER has more than one PLATFORM location and the TACTICAL CRITERION includes a condition on the relative geometry of the subordinate and the target. For example, if a commander is evaluating whether to assign a target to a subordinate, the decision may be based on the distance from the subordinate to the target. Only one location of the subordinate is used to calculate this distance and that location may not even have the components capable of engaging the target; other locations (that may or may not be able to attack the target) are not considered.
Platform 6.3 Logic Processes	• Track correlation is perfect and data fusion is accurate, but may be incomplete.	• Unrealistically accurate target data may lead to unrealistic target attrition.
	• Track discrimination is perfect.	• There are no misclassification effects.
Environment 1.0 Atmospheric Characteristics	• The transmission medium does not produce ducting effects.	• Signal transmission may be unrealistic, especially for underwater signals.
	• Refraction effects are based upon a user-defined effective earth radius.	• Variations in the transmission medium do not produce variations in refraction effects. Refraction is not modeled for signals that travel at altitudes less than zero.

TABLE 3-1. Assumptions for SWEG 6.5.5 by Functional Element. (Contd.)

Functional Element	**Assumption**	**Implications for Model Use**
Environment 2.0 Topographic Characteristics	• Distances are based upon a projection of a spherical earth onto a flat plane. The radius of the earth is set to 6371221.3 meters.	• For reasonably accurate distance measurements, scenarios may cover at most 80% of any hemisphere.
	• The elevation of the surface of the earth is represented by a continuous, but not necessarily differentiable, function of x and y.	• Since the surface has only one altitude at any x-y location, some terrain features (caves, cliff overhangs, etc.) can not be accurately represented. (This is also an assumption in DMA terrain data). Terrain-following or terrain-avoiding aircraft movement is fairly well simulated due to the surface continuity, but movement on the earth surface (for tanks, trucks, etc.) may be unrealistic due to sharp angles for hills or valleys.
	• Clutter signal may be produced only by a terrain patch directly below the target.	• S/I may be unrealistically high for some systems in some terrain conditions.
	• Multipath and diffraction are not explicitly modeled.	• Since these are not explicitly modeled, signal strength may be unrealistically high or low in some cases.
Command, Control, and Communications 1.0 Command Chain Hierarchy	• No assumptions are made about the organization and relationships of forces.	• The user has the responsibility and the freedom to define all aspects of the command structure, including reactions to commands.
Command, Control, and Communications 2.0 Network Communication	• Messages of the same priority are transmitted on a FIFO basis.	• No interruptions are explicitly represented.
	• Net transmission capabilities are limited only by the user-defined time delays for message types sent over the net and the fact that only one message at a time is sent.	• Multiplexing is not explicitly represented.
	• Messages are either received in total or not at all.	• Packet-switching networks are not explicitly represented.

3.2 LIMITATIONS

Table 3-2 lists known model limitations and assesses the implications for model use. Model limitations are discussed in greater detail in Section 2 of ASP-II. New limitations may be uncovered as further analysis is conducted. Similarly, as enhancements are added to new versions of SWEG, some of these limitations may be eliminated.

TABLE 3-2. Limitations for SWEG Version 6.5.5 by Functional Element.

Functional Element	**Limitation**	**Implications for Model Use**
General	• An external asset must have one of 20 names.	• These names were defined for specific assets at ACETEF, but they could be used as names for any assets. The only asset names that have some assumptions built into the code are MASTER MODEL, ATEWES, and ETEWES.
	• Status messages are not always sent back to another asset when SWEG cannot respond as expected to a stimulus.	• This could cause system crashes or infinite looping in other assets that have not been notified that the conditions in the exercise do not meet the expectations normally assumed for the message that the asset sent to SWEG.
	• The size of addresses is 32 bits.	• This limits the number of PLAYERs, the amount of terrain, and the time in an exercise. The limit on the number of PLAYERs depends on the complexity of the exercise; e.g., a very simple exercise could have two million PLAYERs, while a scenario like the final_battle has a limit of about 137,000 PLAYERs. The amount of terrain is limited to about one million square kilometers.
	• During an exercise, all information is kept in volatile memory (not on a disk).	• The maximum size and complexity of a scenario is limited by the addressable memory size of the hardware.
	• All arithmetic (floating point and integer) uses 32 bits.	• This limits the relative sizes of the smallest possible time increment and the total length of the exercise; e.g., for a two hour exercise, the smallest time step is about one millisecond, while a precision of one second will allow for exercises up to about three months. This also limits the accuracy of real arithmetic results.
	• Semantic codes for each type of entity have six (decimal) digits.	• Since two of these digits are reserved to indicate the category (PLAYER, PLATFORM, ELEMENT, etc.), each category is limited to 10,000 types each.
	• Contents of internal memory may not be saved if an exercise is unexpectedly stopped.	• This hinders analysis. It primarily affects virtual exercises since stopping in the middle of a run is the usual practice in that mode.
	• External interface events are treated like any other events in an exercise; thus they can affect the results of the exercise.	• This includes interface with external assets and interface with the user via textual or graphical data capture. In constructive mode, for example, running an exercise with graphics can give different results than running the same scenario without graphics.

TABLE 3-2. Limitations for SWEG Version 6.5.5 by Functional Element. (Contd.)

Functional Element	**Limitation**	**Implications for Model Use**
Platform 1.1 Configuration	• Aggregation changes are limited to changing an expendable into a PLAYER with a single PLATFORM or changing a PLATFORM into an expendable item belonging to a thinker.	• This prevents the representation of activities such as those that result in the improvement of a PLAYERs own status. For example, a PLAYER cannot rebuild or fix a radar or building or runway. New supplies coming into a PLAYER are absorbed by a thinker and cannot be transferred to another system within the PLAYER.
Platform 1.2 Movement	• There is no explicit capability for multiple PLAYERs to move in formation.	• This capability is explicitly provided for multiple PLATFORMs of a single PLAYER, but if the user wants separate perceptions, maneuver capabilities, etc., for each platform in a formation, this capability is inadequate.
Platform 1.4 Vulnerability	• There is no explicit instruction for defining collateral damage.	• Immediate effects of collateral damage can be represented by using weapons that can simultaneously attack platforms near the point of detonation; however, time-varying effects cannot be easily represented.
Platform 2.0 Sensors	• Explicit S/I calculations must be performed to represent sensor operations.	• Implicit representation of detection would be useful for users who care only about relative geometry (line of sight, range, altitude, azimuth, and elevation) of the sensor and the target. Energy transmission for jammers and communications can be implicitly represented.
Platform 3.0 Weapons	• No reason for an aborted shot is provided. • Weapons cannot fire at a location unless there is a perceived PLATFORM at that location.	• This primarily affects other assets in a virtual exercise. • There is no explicit representation of weapons which can launch ordnance at a set of coordinates and then detonate.
Platform 4.0 Communication Devices	• Communication transmitters and receivers must always be defined in pairs.	• In order to send or receive messages, a PLAYER must have both devices. This is different from world reality and different from the representation of sensors in SWEG, where only a receiver is necessary for detection.
Platform 6.1 Decision-Making Capabilities	• Resource allocations instructions are limited to nine categories; absorb, communication method selection, emission control, intel, lethal assignment, lethal engagement, maneuver, nonlethal engagement, and request.	• These categories represent most capabilities that a user would want; however, flexibility is somewhat limited.

TABLE 3-2. Limitations for SWEG Version 6.5.5 by Functional Element. (Contd.)

Functional Element	**Limitation**	**Implications for Model Use**
Platform 6.2 Decision-Making Logic	• Resource allocation is limited to evaluating multiple resources for possible allocation to one target.	• There are cases where it would be more realistic to evaluate multiple targets for one resource. For example, once a queue of targets has been established for each of several weapons, it would then be easier to pick the "best" target from the queue for each weapon, rather than continuing to examine each individual target against all of the weapons.
Environment 1.0 Atmospheric Characteristics	• Meteorological effects cannot be represented except for completely opaque clouds or globally applicable weather.	• Weather can be represented in the transmission loss tables, but this does not allow for the localized effects that are most often desired.
Command, Control, and Communications 2.0 Network Communication	• Messages are required to include information, not just an action request.	• Even though the messages are user-defined, the format requires that they include some information. This usually is not a problem, since most world reality messages with actions to be performed also contain some information.

3.3 ERRORS

Errors are conditions in which a correct model input set will cause the model to abort or to give incorrect results. Table 3-3 lists known errors in SWEG Version 6.5.5, and assesses the implications for model use. Not all of the errors listed in Table 3-3 have been formally documented as SWEG SCRs within the ACETEF CM process. The list of ACETEF SCRs is presented in Appendix B.

It is important to note that the table below includes only errors which affect the functionality of the model from the user's point of view. Instances of non-standard, inefficient, or unused code are not listed. This could be a problem for experienced SWEG users who are accustomed to looking in the bootstrap file to determine which instructions are implemented; there are several instances of instructions included there which do not work correctly. Some of these have worked correctly in the past, and some may do so in the future, but do not now work. These are listed as errors only if they are included as instructions in the SWEG User's Guide.

Compiler warnings are errors from a software designer's viewpoint, but not from a user's viewpoint. Thus they are not included in the table unless they indicate an error that affects the user. SWEG uses exception-handling, which some C++ compilers do not allow; however, there is an option to turn off exception-handling, so this does not lead to errors.

TABLE 3-3. Model Errors in SWEG Version 6.5.5 by Functional Element.

Functional Element	**Error**	**Implications for Model Use**
General	• Platforms are sometimes displayed by the graphics after they have been removed from the exercise.	• This can be confusing, even though only the location, not the icon, is shown.
	• The latitude/longitude graphics option does not work.	• No grid at all is displayed if this option of the AXES instruction is chosen.
	• Terrain contours plotted with graphics always have straight left, right, top, and bottom edges.	• The edges should approximate arcs of ellipses. Testing has determined that the translation from latitude/longitude to x-y coordinates works correctly. It is not clear if the plotting is merely cut off before the actual edges are reached, or if there is an error in the plotting logic.
	• The "UPDATE-EVERY" instruction for graphics situations does not work correctly.	• If this instruction is defined for one situation, it will also be applied to situations for which it is not defined. This causes a tremendous inefficiency in the graphics display and causes many unnecessary events to be processed. This can hamper the use of graphics during real-time exercises.
	• The model responses, ON-ITS-OWN, ENGAGES and, BY-COMMAND, ENGAGES are not necessarily correct.	• Either of these may be produced in either case. The incorrect listing can be confusing and affect analysis results.
	• Other incidents are not correctly reported in the listing file are: - CAN'T-START-LETHAL-ENGAGE-OF - CAN'T-USE-NEW-DETECTION-OF - CRASHES-INTO-THE-GROUND - DROPS-MANEUVER-QUEUE-ENTRY - HAD-A-BAD-LAUNCH-AGAINST - LOST-LAST-SUB-TO-ATTACK - USING-TRACKER-FOR-SEARCH-ALSO	• The incorrect listings can cause confusion and affect the analysis results.
	• Geometry values are incorrectly printed for at least one incident.	• An incorrect target-sensor bearing is written to the listing file in the FIRST-HAS-SENSOR-IN-RANGE-OF incident. The other bearing might also be incorrect, and it is possible that some other incidents are also incorrectly reporting information since common code is used across incidents.
	• Player creation incidents give the wrong reason for creation.	• DYNAMIC EXTERNAL REQUEST is printed instead of SELF GENERATED
	• The PATH listing option in the ADB instructions does not work correctly.	• If used, the distance reported for the last path point is an erroneous, very large integer.

TABLE 3-3. Model Errors in SWEG Version 6.5.5 by Functional Element. (Contd.)

Functional Element	**Error**	**Implications for Model Use**
General (Contd.)	• Paths are incorrectly printed during the RDB or CDB step or during the ADB step with the time-window option.	• In the ADB, the time window instruction is ignored and the whole path is printed. During the RDB or CDB step, the entire path is printed when the PRINT PATH debug flag is turned on. Both of these errors result from a single error in the code.
	• There is no check in the SDB step to ensure that PLATFORM names match PLATFORM names in the TDB step	• If they do not match, no errors are generated during the SDB, but later steps end prematurely with a core dump.
	• Whenever a resource allocation option has more than one path leading to a SELECT instruction, an error check may incorrectly diagnose a user error.	• The error is reported whenever a path after the first path has a filter with a different number than any filter in the first path. This must occur, for example, if there are more filters in the second path than the first. A user who knows this can work around it by inserting dummy filters in the first path, but otherwise, execution is stopped with a semantic error diagnosis from SWEG.
	• PLAYERs cannot begin existence after the start of the scenario unless they are disaggregated from another PLAYER.	• For constructive mode runs, this is merely a limitation, but in a networked virtual mode, errors can arise when PLAYERs are detected before they are supposed to enter the scenario.
Platform 1.1 Configuration	• Shape data is not correctly shared among PLATFORMs of the same type.	• During the TDB PLAYER structure semantic processing, no dummy PLATFORM is created for the PLATFORM(s) defined in the data item; the dummy should be created to store the SHAPE of the PLATFORM. The potential effects of this are unknown, but since SHAPEs are used only for their masking effects, there may be some errors in masking.
Platform 1.2 Movement	• Turn limits defined by maximum G's or by limits on roll, pitch and yaw do not work correctly.	• Turn limits can be defined only by turn radii, but this is not documented. Use of the other definitions could be confusing and affect analysis results.
	• Single PLAYER (multiple PLATFORM) formation movement does not work correctly.	• Since this has not been thoroughly tested, the scope of this error is unknown.
	• Pitch maneuvers can cause the yaw angle to be calculated incorrectly.	• The platform orientation may be incorrectly calculated if there is a pitch-up or pitch-down maneuver. The error will show up as an incorrect yaw (or heading) angle. This causes confusion and may affect analysis results.

TABLE 3-3. Model Errors in SWEG Version 6.5.5 by Functional Element. (Contd.)

Functional Element	Error	Implications for Model Use
Platform 1.2 Movement (Contd.)	• SWEG decides two points are distinct whenever the horizontal distance between them is zero.	• Completely vertical paths cause several errors in orientation and path creation. Errors in orientation can cause errors in lookup of signature, Pk, etc.
	• Threat or terrain avoidance causes an incorrect initial orientation. The initial vector is incorrectly calculated based on an assumption in the code that the first point on the path has a zero elevation.	• The error shows up only when the PLATFORM does not start moving immediately. Errors in orientation can cause errors in lookup of signature, Pk, etc.
	• Threat avoidance does not always work for simple cases; e.g., when the initial path is parallel to either internal Cartesian axis and passes directly over the threat.	• If the path is changed so that it is not parallel to an axis, the PLATFORM does maneuver to avoid the threat.
	• Threat avoidance tactics may not lead to avoidance of the correct threat area.	• The model assumes that the area to be avoided is the region covered by the threat sensor, and assumes that the height of the sensor antenna is three meters above the PLATFORM location. This may be incorrect.
	• Orientation is sometimes calculated incorrectly at the end of a path. The error will sometimes manifest itself as an extra arc of a circle at the end of a missile flyout path.	• Errors in orientation can cause errors in lookup of signature, Pk, etc.
	• The unit velocity vector is incorrectly calculated for some uses; the velocity at the beginning of a line segment is used throughout the line segment.	• This is especially a problem when a turn is involved. Errors in orientation can cause errors in lookup of signature, Pk, etc.
	• A stop movement instruction may not work correctly.	• In some simple test cases, a player instructed to stop after the first point in its scripted movement path does not, in fact, stop there.
Platform 2.0 Sensors	• SWEG changes the character strings associated with an ELINT code if more than one of the same type exist.	• This affects the CDB step only. It causes an incorrect interchange between SWEG and one of the radio frequency emitting assets.
	• There are several small errors in the sensing code.	• The exact effect of these is unknown; they were discovered during changes to another SWEG project (Turquoise) and were not fixed in Diamond as per government instructions.
	• S/I ratio in the presence of jamming is incorrectly calculated when the ONE-M2-DETECT-RNG data item is used.	• Burn-through ranges with standoff jamming will be incorrect.

TABLE 3-3. Model Errors in SWEG Version 6.5.5 by Functional Element. (Contd.)

Functional Element	**Error**	**Implications for Model Use**
Platform 3.0 Weapons	• If the parent of a newly-created PLAYER is moving, it is assumed that there is a target for the parent and the PLAYER.	• This causes crashes during the RDB or CDB step whenever a moving weapon disaggregates another PLAYER, e.g., if an aircraft launches a missile. This error does not currently occur because it is masked by another error in logic processing.
	• If a weapon has multiple types of expendables, and an expendable of a type that is not listed first is needed, the first type is used instead of the correct type.	• The weapon may use the wrong ordnance for a specific type of target.
Platform 4.0 Communication Devices	• SWEG changes the character strings associated with an ELINT code if more than one of the same type exist.	• This affects the CDB step only. It causes an incorrect interchange between SWEG and one of the radio frequency emitting assets.
	• There is no check for the existence of a communication antenna before trying to use one. An antenna does not exist if communication is modeled implicitly.	• This could cause crashes during an RDB or CDB step.
Platform 5.0 CM/CCM	• SWEG changes the character strings associated with an ELINT code if more than one of the same type exist.	• This affects the CDB step only. It causes an incorrect interchange between SWEG and one of the radio frequency emitting assets.
	• If a disruptor has multiple types of expendables, and an expendable of a type that is not listed first is needed, the first type is used instead of the correct type.	• The disruptor may use the wrong type of expendable for a specific type of target.
Platform 6.2 Knowledge Base	• The message interpretation function stores the sender's local track ID in the local track ID address.	• The sender's local track ID is also stored in the correct place, but this can cause problems if the recipient has a perception with this local track ID, or gets one before the perception from the message is dropped. This error can cause PLAYERs to perform an action that would not otherwise occur.
	• Perceived information is incorrectly updated based on messages sent from one PLAYER to another.	• This can cause PLAYERs to think they have engaged a target when they have not done so, and prevent commanders from making assignments. It causes message recipients to have the same internal status as the message sender, which could cause many different types of incorrect results.
	• Sensor-derived perceptions of itself do not work correctly for a disaggregated PLAYER.	• This prevents it from being able to maneuver without a separate target.

TABLE 3-3. Model Errors in SWEG Version 6.5.5 by Functional Element. (Contd.)

Functional Element	**Error**	**Implications for Model Use**
Platform 6.2 Knowledge Base (Contd.)	• The materiel status is not initialized correctly for a disaggregated PLAYER with a friendly perception of itself.	• Resource allocation and requests will not work correctly for the disaggregated PLAYER.
Platform 6.3 Logic Processes	• The tactical criterion PENDING REQUEST does not work at all.	• In fact, there is no vocabulary entry for it in the bootstrap file. This prevents the user from employing this criterion.
	• The tactical criterion FILL REQUEST does not work correctly in resource allocation.	• The code that processes the tactical criteria for active allocation options does not have a case for that criterion, so it should not be used. This error currently masks the following error.
	• Even if the previous error were fixed, requests generated by a PLAYER for itself do not work correctly.	• This prevents many aspects of logistics from working correctly. It also prevents PLAYERs from requesting air support, artillery support, etc. This error currently masks the following error.
	• If the parent of a newly-created PLAYER is moving, the model assumes that there is a target for the parent and the new PLAYER.	• This error causes crashes during the RDB or CDB step whenever it occurs; however, it currently does not occur since it is masked by the previous error.
	• If a system has multiple types of expendables, and an expendable of a type that is not listed first is needed, the first type is used instead of the correct type.	• This can cause the wrong type of expendable to be decremented and it can cause or prevent PLAYER creation when the opposite should happen.
Command, Control, and Communications 2.0 Network Communication	• There is no check for the existence of a perceived communication net before using the net, even though the specific communication device might not be on a net.	• This could cause crashes during an RDB or CDB step.
	• There is no check on whether or not a specific message type can be sent over a specific net before trying to send one.	• This could cause crashes during an RDB or CDB step.
	• The user cannot specify that a message type include two important types of information: ALIVE/DEAD status and PERCEIVED SIDE.	• This can cause perceptions to be kept when they should be dropped, and prevent engagements from occurring.

Additional possible errors have been found in SWEG using a lint tool. Approximately 15,000 messages were originally produced by the tool, but most of these have been eliminated as possible functional errors. However, about 1,000 message need to be examined in detail to determine whether or not they indicate functional errors. That type of examination was beyond the scope of this task.

3.4 IMPLICATIONS FOR MODEL USE

SWEG is a general purpose conflict simulation, designed for use in a wide variety of applications in both constructive (i.e. stand-alone) and distributed (or virtual) environments. There are no restrictions on the type of platforms that can be modeled, and the only restrictions in platform functions are those imposed by the SCL.

In constructive applications, SWEG is most commonly used for mission- or raid-level analysis. SWEG was not designed to model detailed, engagement-level combat effectiveness, and does not explicitly represent details of sensor or weapon system performance such as dynamic signature fluctuations, detailed signal processing and tracking errors or deceptive ECM, detailed environmental characteristics (such as terrain clutter, multipath, and diffraction), or detailed weapon guidance and miss distance, which are usually required for engagement analysis. SWEG has some capability to model logistic resupply and has been used in multi-day scenarios; however, there are other limitations, such as no maintenance modeling or automatic sortie generation, that would limit its usefulness in campaign analysis.

Of the assumptions and limitations listed in Tables 3-1 and 3-2, relatively few are significant for constructive, mission-level analysis, and there are often "work-arounds" for those that are significant. Two of the more significant model limitations for mission-level analysis are the assumption of perfect sensor data fusion and correlation and the lack of redundancy or backup in the command and control hierarchy. The list of errors in Table 3-3 and in Appendix C seems extensive and may be a consequence of the fact that SWEG has become available only recently to users outside of the ACETEF and is now being used for different types of analysis applications.